Exploring the Potential of Combining Smart Glasses and Consumer-grade EEG/EMG Headsets for Controlling IoT Appliances in the Smart Home

E. De Buyser, E. De Coninck, B. Dhoedt, P. Simoens

*Ghent University - iMinds, Technologiepark 15, B-9052 Gent, Belgium

Keywords: smart glasses, EEG/EMG headset, smart home

Abstract

The number of smart home appliances that can be connected to the Internet grows every day. In this paper, we explore the potential of combining two emerging head-mounted interaction devices for intuitive control of these devices. Smart glasses are used to detect the object the user wants to control, and an EEG/EMG headset is used for triggering commands to the object of interest. We discuss the research and implementation challenges of identifying devices having the users visual attention and of mapping EEG/EMG headset output to device instructions. By exploiting the user context, we improve the responsiveness and precision of the user intent detection. Despite the enthusiastic reactions of the participants in a small user study, we have learned that the consumer-grade headsets available today present many shortcomings.

1 Introduction

With the aging population, more elderly with impaired mobility will stay longer at home. Being able to activate household appliances without physically displacing oneself is thus of great interest to this population. An increasing number of household appliances is being connected to the Internet; transforming our domestic environments into remotely controllable smart homes. Thermostats, light switches and LED lights are examples of Internet-of-Things (IoT) devices already commercially available, many more devices will follow suit.

Today, the mainstream interaction pattern to control these smart home appliances is based on apps. As each device comes with its own companion app, this is a rather tedious and awkward approach. For every device control operation, even everyday tasks such as switching on the light, the smart home resident needs to find the appropriate app on his smartphone, launch it and navigate through the vendor-specific user interface. The expected increase of the number of controllable smart home appliances will exacerbate this problem. Elderly and/or impaired users will benefit from a more intuitive interface for control of IoT devices. The recently proposed interaction model with voice commands [1], uttered to a smartphone or specific device (e.g. Amazon Echo), might not be suited for older adults with weaker voices, unrecognizable dialects, etc.

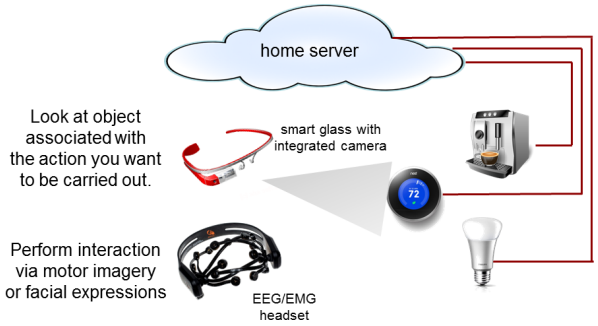

Figure 1. Facial expressions are captured by an EEG/EMG neuro-headset and translated into commands for IoT appliances. The appliance of interest is visually identified by a first person viewpoint camera.

In this article, we present our efforts to build a multimodal assistive system for IoT interaction in the smart home, combining the input of two types of emerging head-worn interaction devices: smart glasses and neuro-headsets. The principle is illustrated in Figure 1.

Capitalizing on the idea that humans have a natural tendency to look at the object they are manipulating or the person they are talking to [2], we use the front camera on smart glasses to capture the user gaze and apply computer vision techniques on this first-person viewpoint video to determine which of the surrounding IoT devices has the users visual attention. The system automatically detects the object in the user gaze and projects the possible device commands in the near-to-eye display of the glass. Instead of having to launch different apps, users can thus just look at the object they want to interact with, or to an object related to it, e.g. the light switch to turn on a light, or the thermostat to increase the heating.

To trigger the desired command for the object being looked at, we use the eMotiv EPOC neuro-headset [3]. This headset has 14 sensors measuring the electric potentials along the scalp that result from brain activity (EEG) and from facial musculature (EMG) when smiling, frowning, etc. The EPOC comes with a (black-box) signal processing SDK that maps the raw

EEG/EMG sensor data onto three categories of events:

- affective: emotions like excitement, frustration, engagement
- cognitive: motor commands like push, pull, rotate, etc.
- expressive: facial expressions like blink, smile, left/right wink, etc.

Our initial goal was to use the cognitive events of this SDK, because humans are used to manipulate objects with motor commands: turning knobs; flipping switches or pushing buttons. The intended users of our system, with limited mobility, would then be able to increase the room temperature by looking at the thermostat from a distance and cognitively rotating it. However, our initial experiments with the motor cognition SDK of the eMotiv EPOC neuro-headset revealed a disappointing recognition precision, even after following the recommended training procedure. As a fallback solution, we resorted to the more robust facial expression functionality of the provided SDK.

In a previous position paper [4], we outlined our vision on building an intuitive interaction system combining brain-computer interfaces and smart glasses. In the present paper, we report on the implementation challenges and performance bottlenecks we observed when building a prototype system using commercially available devices and SDKs. Moreover, we extend our system with context-awareness to support device and command recognition.

The remainder of this paper is structured as follows. In section , we explore related work on using smart glasses and brain-computer interfaces in the domain of smart homes. In section 3, we introduce the software architecture of our system. In section 4 we elaborate on our video processing pipeline for object recognition, in section 5 we leverage on context information to improve this process. In section 6, we discuss our device control mechanism. In section 7, we evaluate our prototype, including results of a user study. In section 8, we conclude this paper by presenting our perspectives on technological advancements and alternatives that can be used to further improve the presented prototype.

2 Related work

While both wearable cameras and brain-computer interfaces (BCI) have been extensively studied in the context of smart homes, to our knowledge, we are the first to explore the potential of combining these devices.

The near-to-eye display of smart glasses can be used to present information assisting in the execution of everyday tasks. A comparative study in guidance for kitchen tasks, reported in [5], concluded that smart glasses were indeed one of the most preferred options by the participants. Gabriel [6] is a supporting framework for such wearable cognitive assistance. An alternative to the hybrid BCI used in this paper, is to recognize hand gestures in the smart glass camera feed. Recognizing hand-related activities by a wearable camera is challenging due to the many temporal and spatial variations of hand interactions [7].

P300-based BCI have been used as well to control objects present in our daily life, e.g. smartphones and wheelchairs.

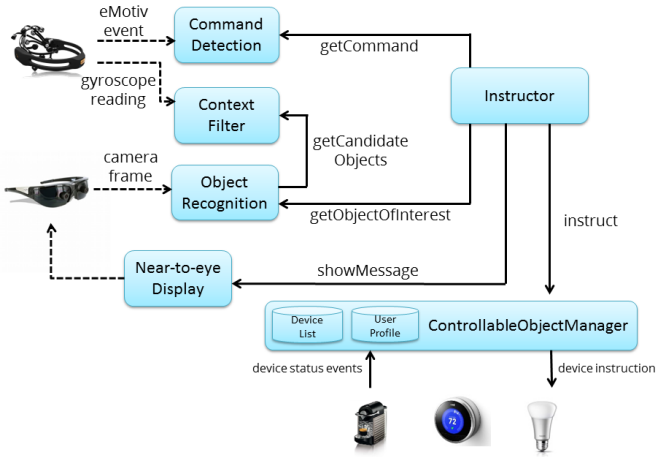

Figure 2. System architecture

The P300 potential is a peak in the EEG signal that is elicited by presenting a series of visual stimuli (e.g. flashing images) of which only one is related to the user's intention. Although these stimuli could be rendered in near-to-eye displays of smart glasses, we believe a P300 interface would hinder user acceptance. Instead, we use the spatial distribution of the EEG and EMG signals along the scalp related to motor imagery and facial expressions, which can be detected without applying external stimuli. The same principle has been used to control a quadcopter [8]. In [9] and [10], the eMotiv headset is combined with an eye-tracker into a system for cursor manipulation and manipulation of a smart TV respectively. By combining a neuro headset with smart glasses, we allow users to quickly change control of one device to another, each time presenting the appropriate instructions in the near-to-eye display of the smart glass and translating the EEG/EMG signals to the correct device control command. Moreover, we introduce context-awareness based on the neuro-headset gyroscope to further improve device recognition.

3 Architecture

The different building blocks and their interaction are illustrated in Figure 3. The smart glass camera feed is processed by the `Object Recognition` module; which detects the objects having the users visual attention. This module gets a list of possible objects from a `Context Filter`; which prefilters the subset of controllable devices present in the smart home that is likely in the users view, based on input such as head position information derived from the gyroscope of the neuro-headset. The `Command Detection` (CD) module maps raw sensor readings to motor commands or facial expressions. In our current implementation, this component is a wrapper around the eMotiv SDK.

The `Instructor` implements the heart of our system: once the object of interest is recognized, it fetches a user command from the CD and translates these to specific device control instructions like turn light on, increase temperature by 10, etc. The `Instructor` also sends corresponding feedback messages to the near-to-eye display of the smart glass, e.g. ask-

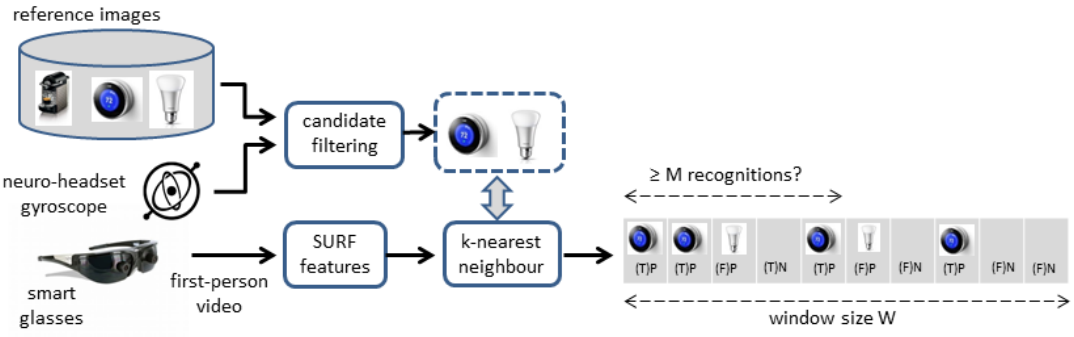

Figure 3. Object recognition pipeline. SURF features are matched against a subset of the objects in the database that is derived from the user gaze orientation. The decision on user intention is taken by deliberation of the number of object recognitions over a window of frames. Object recognitions can be true or false positives (TP/FP). Frames where no objects are detected are either true or false negatives (TN/FN).

ing the user to confirm the recognized object, or listing the possible commands.

The `Controllable Object Manager` (COM) provides the communication to each object in the smart home. It translates the generic device instructions from the `Instructor` into the appropriate, vendor-specific syntax. Internally, it maintains metadata and status information on all controllable smart home devices, as well as a `User Profile` database with the cognitive and facial commands that are most easily generated by the user. Moreover, the COM advertises a mapping between these input commands and possible device instructions to the `Instructor`. This list is adjusted to the state of the devices and allows to tailor the instructions shown to the user. For example, if a lamp is on, the user should only be shown a command to turn it off.

4 Visual Device Identification

To identify the object the user intends to control, we perform object recognition on the frames captured by the forward facing camera of the smart glass, as illustrated in Figure 2.

We follow the traditional approach of matching feature descriptors, calculated on the video frames, with a database of reference images. Arguably, IoT device manufacturers can easily deliver a reference image of good quality. In fact, in many cases useful pictures are already available: the commercial photographs used in web shop catalogs are perfectly suited, since they show the object isolated on a neutral background. At best, we would expect the IoT manufacturer to provide photographs from different angles, although our experiments already give good results with only one reference image per device. We also considered QR codes, whose distinct features might be easier to detect. QR stickers are likely to be kept small since they are visually unpleasing. In our tests with 4.5 x 4.5 cm QR codes and different off-the-shelf recognition apps in the Google Play store, markers were only recognized within perpendicular distances of less than 60 cm and viewing angles up to 40 degrees.

Feature selection is a non-trivial task, especially for the analytics of first-person video, which is characterized by highly dynamic changes and scene characteristics. We opted for SURF features, which is a common choice for object identification in first-person video, according to the survey in [11]. We compare SURF features on each frame captured by the smart glass camera with the (pre-calculated) features of a subset of the images in the database. Features are matched using k-nearest neighbor matching: for each feature in the camera frame (the query features), we calculate the distance to the two (k=2) closest features in the reference image. A feature match is assumed when the difference between the distances of the query feature to the two closest features in the reference image is at least 60 %.

We assume an object has the user's visual attention if it is recognized at least M times in a window of W frames. Our intuition is that users will fix their gaze onto the object of interest. The object can thus be recognized in the majority of the frames in the window considered. To make our system more robust to false positive recognitions, we balance the following three parameters:

- The minimum number of matching features with a reference image before concluding that an object is recognized in a camera frame. A higher threshold decreases the number of false positive recognitions, but increases the number of false negatives. In our prototype, we set the threshold to 1 % of the number of features in the reference image, increased with a static offset of 8 features to avoid too many false positives for objects with fewer features.
- The window size W of considered frames: having a larger window will require processing more frames before a decision is made and thus decrease the responsiveness of the system, but windowing the recognition also mitigates the effect of sporadic false positives. In our current prototype, we use a varying window size between 0 and 10, depending on the number of objects that is likely in the user's view.
- The minimum number of recognitions M in the window required before deciding that the user has the intention to interact with the recognized object. In our current prototype, we use M = 30 %. When this quorum is reached for one object, the remaining frames in the window are skipped. If the threshold M is not reached within the window W, the system returns the object that was recognized the most number of times in the window.

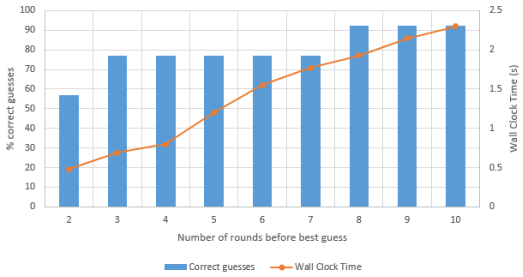

Figure 4. Impact of increasing the size of the window W on the recognition performance with 5 candidate objects.

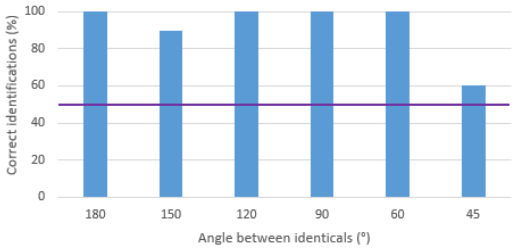

Figure 5. Correct number of identifications for various angles between two identical objects.

Figure 4 shows the percentage of correct recognitions for increasing window sizes W, in a video taken in a typical living room and 5 candidate objects. In the video, the object of interest returned 13 times. The graph also shows the average wall clock time to process an entire window of frames, which can be regarded as a measure of the system's responsiveness.

5 Filtering candidate objects

To recognize an object, the SURF features calculated on the captured frames must be matched to the features of all reference images in the database. Although parallelization of this matching is possible on multi-core processors, our prototype implementation reveals that the major processing bottleneck is not the number of objects in the database, but rather the calculation of features on each video frame. When there is a single object in the IoT database, the entire object recognition pipeline requires 2.25 s, and this increases with only 4 % per additional object in the database.

Despite the above scalability observations, narrowing down the number of candidates is still beneficial since this reduces the probability of recognizing the wrong object and triggering the wrong interaction sequence. Moreover, even a perfect recognition algorithm would not be sufficient when multiple identical IoT devices may be present. Light switches are a canonical example: all switches are visually similar but operate on different light armatures. The framework can thus benefit from additional context information to determine the appropriate light.

We have therefore investigated the use of information provided by other sensors in the neuro-headset to pre-filter the number of candidate objects. We extend the COM module with information about the spatial position of the objects in the room. Object positions could be measured by determining the head position when the user is looking at the object: the azimuth (magnetic direction) measured by a compass and the head tilt measured by a gyroscope (e.g. when looking to a lamp at the ceiling). Lacking a compass in our eMotiv headset, our current prototype only uses the gyroscope for measuring both vertical and horizontal viewing directions. Instead of using the magnetic north as reference point, we measure differential horizontal head rotations once a first object is recognized and assume the user is always standing in the same position to avoid having to take into account the distance from the user to the objects.

To avoid manual configuration by the user, our system self-learns the positions of objects in the room. Initially, none of the object positions is known and all objects are candidates in the object recognition pipeline. User interaction with an object is used as an implicit acknowledgement for correct object recognition and the current looking direction (by reading from the gyroscope) is automatically saved. Spatial object information is thus gradually constructed. The resulting 'virtual floorplan' is then used to exclude all objects that are not likely to be in the user's view. Humans have a horizontal field of view with depth perception of about 114 degrees. Assuming that humans keep the object of interest more or less centered in their gaze, we opt for a slightly smaller range of 45 degrees to the left and right hand side of the looking direction to construct a set of candidate objects.

Narrowing down the subset of candidate devices allows us to reduce the window size W accordingly. If the virtual floorplan indicates that only one device is in the user viewpoint, we can even bypass the entire object recognition pipeline. We scale the window size W proportionally with the number of candidate objects in the dataset until a maximum of $W = 10$.

In order to test the proposed solution, we have placed two identical objects left and right of the user, at varying viewing angles. We performed 20 recognitions, alternating randomly between both objects and generating some random rotation by looking around the room. Figure 5 shows the percentage of correct recognitions for various viewing angles. At 45 degrees, only a slightly better performance than random guessing is obtained, but for larger angles between two identical objects the recognition is almost perfect. The slightly lower accuracy for 150 degrees is due to gyroscope drift and more variations in the head rotation when looking at the object. One would expect to see the same inaccuracy for 180 degrees, however here the objects are aligned exactly left and right to the user, which makes it easier for the user to clearly align his body when looking at the object.

6 IoT Device Control

When the `Instructor` is notified that an object is recognized, it will query the `Command Detection` (CD). The `Command Detection` module filters the stream of events generated by the eMotiv SDK. Each event is reported with a relative strength, indicating the confidence of the SDK, and we

Figure 6. Users get visual feedback in the smart glass display on the possible instructions for the recognized device.

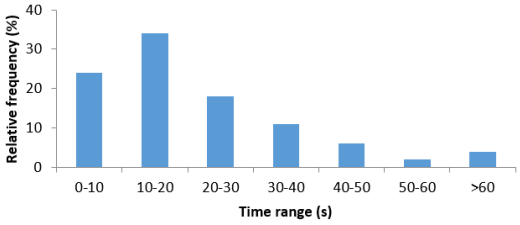

Figure 7. Histogram of interaction times to switch a light on, change its color and switch it off again.

only accept events with a strength of at least 0.3. Moreover, owing to involuntary muscle contractions and low sensor signal quality, we observed that the eMotiv SDK generates events that do not correspond to the users intention. Therefore, the CD module only considers the list of events provided by the `Instructor`. For example, if the `Smile` facial expression is not mapped to the device currently in the user's view, then it should not be considered as an intended user command even if it is detected by the SDK with high strength. Moreover, the CD will return the first event in the list of instructions that is detected 5 times with sufficient strength after the query.

After this, the EMG recognition pipeline is stalled for two seconds to avoid users accidentally triggering a second device control message (e.g. because they keep smiling). The EMG events are mapped onto commands for the IoT device identified through object recognition. To cope with the different protocols and interfaces of IoT devices, we leverage on the our Dynamic Adaptive Management of Networks and Devices (DYAMAND) middleware [9]. DYAMAND abstracts device-specific syntax into service types that describe device functionality in more generic ways. Each service type comes with its own interface and one device can have multiple service types. An example service type is a light service: DYAMAND offers generic methods for turning the light on and off and translates these instructions to device-specific protocol messages. We have developed a novel DYAMAND plug-in that maps service type methods into methods with binary input: an EMG event can indeed only be used to confirm or deny an option, or to choose between two options. For example, the Lamp service type provides a method to configure the brightness level or set the specific lamp color. This is converted to binary user instructions like increase brightness with 25 % or change the color . The list of possible methods is automatically adjusted to the device state: e.g. if the device is on, only a device off method is available.

Visual feedback on the recognized object and the possible instructions are presented in the near-to-eye display of the smart glass. Figure 6 shows the instruction sequence for manipulating the Philips Hue, an IoT-enabled lamp of which the color and the brightness can be configured. When an object is recognized, we display a message asking the user to confirm the detected object. After confirmation, we present a list of possible instructions. This list makes the correlation between the facial expressions and device commands explicit. Arguably, the correspondence with mental motor (EEG) commands and device instructions would be much higher and in this case more intuitive icons could be used.

7 Prototype evaluation

To evaluate our system prototype, we recruited 4 men and 3 women with ages ranging from 19 to 49 years and varying levels of technological literacy. The volunteers were invited to the iMinds HomeLab [12], a semi-realistic home setting where we placed the Philips Hue and two other objects.

Our prototype consisted of the Vuzix 920 AR smart glass and the EEG/EMG Emotiv EPOC headset. Because none of the headsets have a wireless interface nor are their SDKs supporting mobile operating systems, both devices were connected to a laptop (Intel Core 2 Duo 2.26 GHz, 4 GB, Ubuntu 14.04) mimicking the functionality of the users smartphone.

At the beginning of the test, users were asked to try out the facial expressions of the expressive suite of the eMotiv SDK while sitting in front of a tool of the eMotiv SDK that translates the signals picked up by the headset into an avatar mimicking the users facial expression. After these 10 minutes, each participant was asked to rank the facial expressions according to his perceived ease of detection. The `COM` module maps possible device commands to expressions following this ranking. For example, the user in Figure 6 preferred the 'smile' and 'clench' expression, hence these two expressions are always shown in the display.

After configuring the system with the best expressions for the participant, users were instructed to complete a series of interaction scenarios with a Philips Hue IoT light, such as "switch a light off", "change the lamp color" and "switch the lamp off". The distribution of these interaction times as recorded throughout the user tests is plotted in Figure 7. The results comprise the time needed to complete the object recognition pipeline, to show a confirmation of the recognized device and to select one instruction through a facial expression.

The results indicate a mean of 16 s to switch on a light. These relatively high times are not only caused by processing delays for object recognition and EEG/EMG commands: a major contribution to the total delay stems from our deliberate choice to ask the user to confirm the detected object. Arguably, our system is far from being an equivalent as fast as manually touching a light switch. However, this might be acceptable for elderly living at home with impaired mobility. The participants to our user study found the responsiveness more than acceptable, but we hypothesize a novelty bias because of the use of technologies unknown to most of our users. Faster interaction times are crucial for sustained usage on the long term.

8 Perspectives and conclusion

As an alternative to today's app-based or voice-controlled smart home interaction patterns, this paper reports on the design and implementation of a first prototype combining a neuro-headset and smart glasses. The prototype revealed the design trade-offs, yet many challenging research questions still need to be answered.

Improved accuracy and unobtrusiveness of headsets While people are wearing glasses already and Google Glass was a notorious example of how smart glasses can be aesthetic and light enough to be carried 24/7, the current visual appearance of EEG/EMG headsets (the visible sensors giving them a medical aura) make it much harder to imagine people continuously wearing such headsets. The research to less unobtrusive EEG solutions for mobile cognition is making rapid progress. Miniaturized EEG electrodes integrated in discreet baseball caps and individualized ear pieces have recently been reported [13, 14]. It remains an open question which type of cognition can be captured accurately enough with these devices, as the electrodes only cover a limited region of the scalp.

Improving object recognition Relevant techniques to detect the object of interest fall in two categories: applying more advanced computer vision algorithms and further refining the subset of candidate objects through context and data from other sensors, beyond the compass and gyroscope already introduced in this paper. We refer to [11] for a recent survey on computer vision techniques in wearable camera scenarios. Going beyond RGB, depth cameras have recently been integrated in headsets such as the Oculus Rift and mobile devices like Google Project Tango. Depth cameras may allow for a better spatial positioning and in turn improve the candidate object filtering, and RGB-D object recognition algorithms are available [15]. User gaze estimation can also be refined by eye-trackers. Beyond computer vision, we can leverage on smart home sensors to infer daily usage patterns of smart home appliances [16].

We reckon that our vision can only become an acceptable alternative for the general public when the above hurdles are tackled. In future work, we will include novel deep learning techniques for object recognition, and introduce probabilistic methods to deal with uncertainty, e.g. due to partially occluded objects. In the meantime, we believe that after integrating some of the already available technologies today, a prototype can be realized that could be of benefit to elderly with impaired mobility that are less dexterous with apps and might consider it awkward to talk to devices.

Acknowledgements

Part of this research is funded through the iMinds Everything Connected program.

References

[1] F. Portet *et al.*, "Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects," *Personal and Ubiquitous Computing*, vol. 17, no. 1, 2013.

[2] M. Corbetta *et al.*, "A common network of functional areas for attention and eye movements," *Neuron*, vol. 21, no. 4, pp. 761–773, 1998.

[3] R. Lievesley *et al.*, "The Emotiv EPOC neuroheadset: an inexpensive method of controlling assistive technologies using facial expressions and thoughts?," *Journal of Assistive Technologies*, vol. 5, no. 2, pp. 67–82, 2011.

[4] P. Simoens *et al.*, "Vision: smart home control with head-mounted sensors for vision and brain activity," in *Proceedings of the fifth international workshop on Mobile cloud computing & services*, pp. 29–33, ACM, 2014.

[5] J. Wang *et al.*, "Evaluating Different Types of Prompts in Guiding Kitchen Tasks for People with Traumatic Brain Injury: A Pilot Study," in *Proceedings of the 36th Annual Conference on Rehabilitation Technology*, 2013.

[6] K. Ha *et al.*, "Towards wearable cognitive assistance," in *Proc. of the 12th annual international conference on Mobile systems, applications, and services*, ACM, 2014.

[7] T. Ishihara *et al.*, "Recognizing hand-object interactions in wearable camera videos," in *Image Processing (ICIP), 2015 IEEE International Conference on*, IEEE, 2015.

[8] K. LaFleur *et al.*, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface," *Journal of neural engineering*, vol. 10, no. 4, 2013.

[9] C. Brennan *et al.*, "Promoting autonomy in a smart home environment with a smarter interface," in *Intl Conf of the IEEE Eng in Medicine and Biology Society*, 2015.

[10] M. Kim *et al.*, "Quantitative evaluation of a low-cost noninvasive hybrid interface based on EEG and eye movement," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 2, 2015.

[11] A. Betancourt *et al.*, "The evolution of first person vision methods: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, 2015.

[12] http://ilabt.iminds.be/homelab.

[13] M. G. Bleichner *et al.*, "Exploring miniaturized EEG electrodes for brain-computer interfaces. An EEG you do not see?," *Physiological reports*, vol. 3, no. 4, 2015.

[14] J. L. Park *et al.*, "Making the case for mobile cognition: EEG and sports performance," *Neuroscience & Biobehavioral Reviews*, vol. 52, 2015.

[15] K. Lai *et al.*, "RGB-D object recognition: Features, algorithms, and a large scale benchmark," in *Consumer Depth Cameras for Computer Vision*, Springer, 2013.

[16] N. K. Suryadevara *et al.*, "Forecasting the behavior of an elderly using wireless sensors data in a smart home," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, 2013.